Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3680~3689
e-ISSN: 2548-6861

OCR Engines for License Plate Recognition: A Comparative Study of Tesseract, EasyOCR, PaddleOCR, and TrOCR

Afifah Khaerani Aziz [1]*, Marzuki Pilliang [2]*, Diana Kurniawati [3]**
* Faculty of Science and Technology, Salakanagara University
** Faculty of Law and Business, Salakanaga University
afifah.khaerani@unsaka.ac.id [1], marzuki.pilliang@ieee.org [2], diana.kurniawati@unsaka.ac.id [3]

Article Info

Article history:

Received 2026-07-26
Revised 2026-08-01
Accepted 2026-08-10

Keyword:

Automatic License Plate Recognition,
Deep Learning,
Intelligent Transportation Systems,
Optical Character Recognition,
Vehicle.

ABSTRACT

Automatic License Plate Recognition (ALPR) is a critical component of Intelligent Transportation Systems (ITS); however, the Optical Character Recognition (OCR) stage remains a significant bottleneck when confronting diverse linguistic scripts, complex plate formats, and environmental degradations. This systematic literature review comparatively evaluates four primary OCR engines—Tesseract, EasyOCR, PaddleOCR, and TrOCR—to bridge the gap between constrained local optimizations and globally resilient applications. Adhering to the PRISMA 2020 guidelines, a comprehensive search across six major academic databases spanning January 2020 to May 2026 initially identified 647 records. Following rigorous screening processes, a final core corpus of 21 empirical studies was qualitatively synthesized to account for extreme cross-study hardware and dataset heterogeneity. The analysis reveals that no single engine is universally superior; efficacy is fundamentally dictated by their underlying neural architectures and contextual deployment parameters. Tesseract offers maximum computational efficiency but fails significantly on non-Latin and complex scripts due to legacy segmentation limits. EasyOCR provides an optimal accuracy-to-speed ratio, making it highly suitable for real-time edge device deployments. PaddleOCR excels in robust sequential decoding, delivering the highest exact-match rates required for high-stakes applications like automated tolling. Conversely, the Transformer-based TrOCR emerges as the definitive frontier for highly complex, irregularly spaced, and multilingual plates (e.g., Han, CIS region scripts), though its severe computational latency currently restricts it to cloud-based infrastructures. Furthermore, the synthesis establishes that dynamic, environment-aware preprocessing is a mandatory prerequisite to mitigate visual stressors such as motion blur and low illumination. This review provides a novel, context-driven deployment taxonomy, equipping researchers and developers with actionable guidelines to navigate the trade-offs between architectural accuracy, script diversity, and computational resource constraints. Finally, the review identifies unified Large Vision-Language Models (LVLMs) as the next-generation trajectory to resolve current multi-stage pipeline dependencies.

I. INTRODUCTION

Automatic License Plate Recognition (ALPR) serves as a foundational pillar in the realization of efficient and reliable Intelligent Transportation Systems (ITS). ALPR systems enable the automated identification of vehicles through digital imagery, a capability indispensable for applications such as automated toll gates, smart parking management, electronic traffic law enforcement (e-ticketing), and restricted area security surveillance [1], [2]. Within the ALPR pipeline, the most critical and error-prone stage is Optical Character

Recognition (OCR)—the process of extracting alphanumeric text from the detected license plate images [3], [4].

Driven by rapid advancements in deep learning, various OCR engines have been developed utilizing fundamentally distinct architectural approaches. These range from classical segmentation-based engines like Tesseract, to Convolutional Neural Network (CNN)-based sequence engines such as EasyOCR and PaddleOCR, up to cutting-edge Transformer architectures like TrOCR [5], [6]. Each engine presents unique structural advantages and limitations, particularly when confronted with the vast diversity of plate formats, complex script morphologies (Latin, Bangla, Arabic, Thai, Devanagari, Han), environmental conditions (fog, rain, motion blur), and strict computing constraints (edge devices). Consequently, a profound understanding of the comparative performance of these four OCR architectures is paramount for system developers and policymakers.

Although numerous studies have evaluated OCR performance individually or within country-specific contexts, there is a distinct absence of a systematic review comprehensively comparing these four primary OCR engines in the context of cross-regional and cross-script license plate recognition. Previous studies tend to be restricted to a single nation—such as [7] for Bangladesh, [8] for Indonesia, and [3] for Romania—or only compare two specific engines [9], [10], [11], are heavily oriented toward hardware resource optimization on specific edge nodes (e.g., Raspberry Pi) rather than providing a holistic analysis based on neural architecture, script diversity, and environmental robustness. This gap engenders a practical dilemma: ALPR system developers lack definitive guidelines regarding which OCR engine is most suitable for their specific operational context. Without a systematic mapping framework, the selection of an OCR engine often relies on trial-and-error, significantly impeding the efficient development and global adoption of ALPR technologies.

Diverging from existing reviews within the literature corpus, this study offers substantial scientific novelty. While earlier reviews have evaluated components like YOLO and EasyOCR for integrated rider safety systems [12], or evaluated computational metrics on edge devices [11], they often bypass the deep architectural vulnerabilities of the OCR models themselves. This systematic review transcends these prior studies by integrating the four predominant OCR engines into a single, cohesive analytical framework. It explicitly analyzes performance based on script morphology and challenging visual degradations, synthesizing empirical findings across studies spanning at least ten countries to provide a truly global perspective. Ultimately, this review goes beyond flat accuracy reporting by delivering a novel, multi-dimensional deployment taxonomy that maps optimal OCR engines based on strict hardware requirements, environmental variables, and script complexity.

To systematically address these literature gaps, this review is designed to answer three core Research Questions (RQs). First, RQ1 investigates how Tesseract, EasyOCR, PaddleOCR, and TrOCR compare in ALPR contexts regarding accuracy metrics (such as Character Error Rate) and computational efficiency (including inference latency and memory footprint). Second, RQ2 examines the specific architectural strengths and vulnerabilities of each engine when deployed against highly diverse script types and challenging environmental visual degradations. Finally, RQ3 synthesizes these empirical findings to formulate contextual guidelines for selecting the optimal OCR engine across specific deployment scenarios, ranging from low-power, real-time edge devices to high-fidelity cloud systems.

The principal contribution of this review is the provision of a structured, comparative synthesis that not only reports quantitative metrics but qualitatively elucidates the underlying neural mechanics of why specific OCR engines excel or fail in certain contexts, paving the way for next-generation multimodal architectures. The remainder of this paper is organized as follows: Section II details the PRISMA-guided methodology and extraction parameters. Section III presents the comparative results and discussion, mapped by engine architecture, script, and environmental deployment. Finally, Section IV concludes the paper and proposes an agenda for future research.

II. METHOD

This systematic literature review (SLR) was formulated in strict adherence to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 guidelines to ensure absolute transparency, reproducibility, and methodological rigor [13], [14]. The review protocol was specifically designed to evaluate and compare the performance profiles of four primary Optical Character Recognition (OCR) engines—Tesseract, EasyOCR, PaddleOCR, and TrOCR—within the specific context of Automatic License Plate Recognition (ALPR).

A. Search Strategy and Information Sources

To ensure a highly reproducible discovery phase, a systematic literature search was conducted across six major academic databases: IEEE Xplore, Scopus, Web of Science (WoS), SpringerLink, ScienceDirect (Elsevier), and MDPI. The publication timeframe was strictly restricted from January 2020 to May 2026 to capture the modern era of deep learning advancements, specifically the transition from convolutional networks to Transformer-based OCR architectures. A core Boolean search string was constructed by combining three conceptual blocks representing text extraction, the automotive application, and the targeted architectural models, as follows: *(OCR OR "Optical Character Recognition" OR "text recognition" OR "character recognition") AND ("license plate" OR "number plate" OR "registration plate" OR "ALPR" OR "ANPR") AND ("Tesseract" OR "EasyOCR" OR "PaddleOCR" OR "TrOCR" OR "PP-OCR")*. This core query was executed within the metadata fields of titles, abstracts, and keywords; however, the exact syntax was specifically adapted to the unique search

interfaces and field restrictions of each individual database to maximize retrieval sensitivity while maintaining strict reproducibility.

For IEEE Xplore, the query was applied using the Abstract and Document Title field restrictions, with single quotes enclosing exact phrases to accommodate the platform's search parser. For Scopus, the syntax was converted to the TITLE-ABS-KEY prefix to simultaneously search titles, abstracts, and keywords, which is the platform's recommended approach for comprehensive topic retrieval. For Web of Science, the TS = (Topic Search) prefix was utilized, which automatically maps to titles, abstracts, and author keywords. For SpringerLink, the text: field specifier was employed to cover full-text content, given that this platform indexes a broader range of metadata. Both ScienceDirect (Elsevier) and MDPI utilized the title, abstract, keywords field restriction with appropriate Boolean delimiters (e.g., title, abstract, keywords ((...))) to ensure precise filtering. The initial application of these tailored queries across all six databases yielded a total of 647 records, distributed as follows: IEEE Xplore (n=112), Scopus (n=184), Web of Science (n=78), SpringerLink (n=96), ScienceDirect (n=97), and MDPI (n=80). This multi-database, syntactically adapted approach ensured comprehensive coverage of both engineering-oriented (e.g., IEEE, WoS) and applied-computer-science literature (e.g., Scopus, Springer, Elsevier, MDPI), while the explicit disclosure of per-database syntax and result counts guarantees that the entire discovery phase can be independently replicated by future researchers.

B. Eligibility Criteria

To establish a highly specific and unbiased boundary for study inclusion, a strict set of eligibility criteria was formulated. These criteria evaluate the publication type, language, quality, and the specific characteristics of the ALPR datasets utilized in the primary studies. A primary study was required to satisfy all inclusion parameters while avoiding all exclusion parameters to qualify for the final synthesis, as detailed in Table 1.

TABLE 1.
INCLUSION AND EXCLUSION CRITERIA MATRIX

Code	Type	Description
I1	Inclusion	The study explicitly evaluates or compares at least one of the four specified OCR engines (Tesseract, EasyOCR, PaddleOCR, TrOCR) in an ALPR context.
I2	Inclusion	The study presents measurable quantitative performance data, explicitly reporting standardized metrics such as Character Accuracy, CER, WER, exact-match rate, inference time, or memory consumption.
I3	Inclusion	The research utilizes real-world or high-fidelity synthetic license plate image datasets with thoroughly documented testing conditions (e.g., specific scripts, formats, or environmental noise).
I4	Inclusion	The article is a peer-reviewed academic publication (journal, conference proceeding, or book chapter) written entirely in English.
E1	Exclusion	The study focuses solely on vehicle or plate bounding-box detection without implementing or evaluating the downstream OCR text-extraction algorithms.
E2	Exclusion	The study utilizes the specified OCR engines for non-ALPR purposes, such as general document scanning, natural scene text, traffic signs, or product packaging.
E3	Exclusion	The study lacks transparent quantitative data, relies on undocumented closed-source APIs, or fails to provide extractable baseline performance metrics.
E4	Exclusion	The research is purely theoretical, proposing high-level conceptual frameworks or policies without direct experimental validation on actual image datasets.

C. Study Selection Process

The study selection protocol was executed independently by two primary reviewers to minimize selection bias, with any localized interpretation discrepancies resolved through formal consensus discussions. The filtering process followed a rigorous three-stage pipeline. The first stage consisted of Title and Abstract Screening to rapidly eliminate articles clearly falling outside the computer vision and ITS domain. The second stage involved Full-Text Retrieval for all records that passed the initial screening or presented ambiguous abstracts. In the third and final stage, a Full-Text Assessment was conducted, where the complete methodologies and result sections of the articles were rigorously evaluated against the predefined eligibility criteria. Studies lacking sufficient OCR performance data or clear methodological architectures were systematically excluded from the final corpus.

D. Data Extraction and Categorization

To facilitate a systematic comparative analysis, a tailored digital extraction form was developed to capture technical and environmental parameters uniformly across all selected studies. The extracted information was categorized into three distinct analytical blocks. First, bibliometric and contextual data were recorded, capturing the authors, publication year, country of dataset origin, specific script types evaluated (e.g., Latin, Bangla, Arabic, Thai, Devanagari, Han, or mixed multiformat), and the presence of environmental testing conditions such as fog, rain, low-light, or motion blur. Second, technical specifications were documented, identifying the exact versions of the evaluated OCR engines, the architecture of antecedent localization models (e.g., YOLO variants or SSD), and the integration of any image preprocessing techniques. Finally, performance metrics were extracted to enable direct computational and accuracy benchmarking. To ensure mathematical consistency during synthesis, these metrics were mapped to the standardized definitions detailed in Table 2.

TABLE 2.
STANDARDIZED OCR EVALUATION METRICS EXTRACTED FROM STUDIES

Metric	Definition	Measurement Formula
Character Error Rate (CER)	Percentage of incorrectly recognized characters relative to the total reference characters.	$CER = \frac{S+D+I}{N} \times 100\%$, where S=substitutions, D=deletions, I=insertions, N=total reference characters.
Word Error Rate (WER)	Percentage of incorrectly recognized words relative to the total reference words.	$WER = \frac{S+D+I}{N} \times 100\%$, calculated at the word unit level.
Character Accuracy (ACC)	Percentage of correctly recognized individual characters.	$ACC = \frac{\text{Correct Characters}}{\text{Total Reference Characters}} \times 100\%$
Exact-Match Rate (EMR)	Percentage of plates where the entire alphanumeric sequence is recognized perfectly without a single error.	$EMR = \frac{\text{Perfectly Recognized Plates}}{\text{Total Tested Plates}} \times 100\%$
Precision and Recall	Measures of exactness and completeness at the character or plate level.	$\text{Precision} = \frac{TP}{TP+FP}$; $\text{Recall} = \frac{TP}{TP+FN}$
Inference Time	Time required for the OCR engine to process a single plate image crop.	Measured in milliseconds (ms) or seconds directly via hardware benchmarking.
Memory Consumption	Amount of RAM/VRAM utilized during inference.	Measured in MB or GB via system profiling on the reported hardware.

E. Comparative Analysis Framework and Synthesis Justification

A fundamental methodological decision was made to utilize a systematic qualitative and thematic synthesis framework rather than a statistical quantitative meta-analysis. A formal meta-analysis requires primary studies to share a high degree of statistical homogeneity. However, the automotive OCR literature exhibits extreme empirical heterogeneity; the reviewed studies evaluated vastly different localized script morphologies, operated under divergent environmental illumination conditions, and utilized non-standardized hardware testing environments ranging from low-power IoT edge nodes to high-performance server GPUs. Consequently, statistically pooling these disparate numerical outcomes into a single meta-analytical effect size would introduce severe confounding variables.

Instead, the synthesis was conducted using a thematic cross-tabulation approach to map architectural performance dynamics transparently. Performance data were first segregated by the specific OCR engine evaluated and subsequently categorized by the script complexity of the tested datasets to identify distinct linguistic dependencies. The robustness of each engine against visual environmental degradations was analyzed as a separate contextual layer. The impact of image enhancement and preprocessing techniques on specific engine architectures was also systematically evaluated. Finally, a deployment taxonomy was formulated using a thematic matrix alignment method. This taxonomy was constructed by cross-referencing the qualitative vulnerabilities of the engines against varying script complexities and their quantitative hardware trade-offs (accuracy versus latency, memory footprint, and GPU requirements), ultimately yielding practical, context-driven deployment recommendations.

F. Study Quality Assessment

To ensure that the primary studies did not introduce structural bias into the synthesis, methodological quality was rigorously appraised using an adapted Joanna Briggs Institute (JBI) Checklist for Quasi-Experimental Studies [15]. This assessment evaluated five critical dimensions of validity: the clarity of the research objectives, the adequacy and completeness of the dataset descriptions, the proper utilization of standard evaluation metrics, experimental reproducibility regarding code or data availability, and the presence of adequate comparative baseline models. Each study was scored across these parameters. Articles scoring low (meeting fewer than 3 out of 5 criteria) were explicitly noted for their methodological limitations during the synthesis, ensuring that their findings were interpreted with appropriate caution regarding their risk of bias [15].

G. Methodological Limitations

This SLR acknowledges several inherent methodological constraints. Primarily, as previously noted, the extreme hardware heterogeneity, varied camera configurations, and distinct dataset characteristics across the primary studies preclude absolute quantitative generalizations and prevent direct head-to-head statistical scaling. Furthermore, because this research constitutes a literature review encompassing the specific timeframe of 2020 to 2026, it does not include direct, single-environment experimental validation where all four OCR engines are tested simultaneously on a unified control dataset. Additionally, the inconsistent reporting of computational metrics across studies meant that certain hardware trade-off comparisons relied on smaller data subsets. Finally, the strict restriction to English-language publications may introduce minor regional biases, meaning the synthesized deployment taxonomy remains fundamentally interpretative and descriptive rather than an absolute mathematical certainty.

III. RESULTS AND DISCUSSION

A. Overview of Studies and Quality Assessment

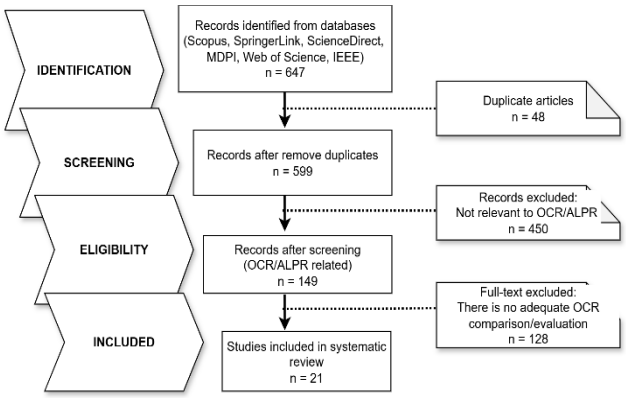

Figure 1. PRISMA flow diagram process

The systematic review process, as visualized in the PRISMA flow diagram (Figure 1), initiated with 647 records retrieved from six academic databases. Following the automated and manual removal of 48 duplicates, 599 unique records underwent initial title and abstract screening. During this phase, 450 records were excluded for lacking relevance to OCR or ALPR applications, leaving an intermediate corpus of 149 articles. A rigorous full-text assessment was subsequently conducted, resulting in the exclusion of a further 128 articles that lacked adequate comparative or quantitative OCR evaluations.

Ultimately, a final core corpus of 21 articles was retained for in-depth qualitative synthesis. Of this corpus, 10 articles were direct experimental comparative studies evaluating at least two engines, while the remaining 11 provided robust quantitative evaluations of a single engine. The synthesized corpus spans diverse geographical regions with varying script complexities, ranging from standardized Latin in Germany [16] and Romania [3], mixed scripts in Indonesia [8], [17] and Malaysia [18], [19], to complex non-Latin scripts such as Bangla, Thai, Devanagari, Arabic, and Han.

Prior to the comparative synthesis, methodological quality was assessed using a modified Joanna Briggs Institute (JBI) Checklist [15]. Overall, 18 of the 21 articles (85.7%) met at least 4 out of 5 criteria, indicating robust methodological quality. An earlier study by [18] met only 3 criteria due to limitations in dataset description and reproducibility. Nonetheless, all articles were retained for their essential quantitative data, with their respective limitations acknowledged during interpretation. The subsequent discussion systematically answers the formulated Research Questions (RQs) by synthesizing findings across the four targeted OCR architectures. The detailed characteristics of the 21 primary studies, including dataset sizes, detection model, and quantitative outcomes, are systematically presented in Appendix A.

B. RQ1: Architectural Deep-Dive and Performance Metrics

Addressing RQ1, the synthesis reveals a stark contrast between classical segmentation-based models and modern deep learning architectures regarding recognition accuracy and computational load. This divergence is fundamentally rooted in the distinct internal neural structures of the evaluated engines.

Tesseract OCR represents the classical paradigm, operating through an analytical pipeline that relies on explicit text-line finding, structural baseline fitting, and polygonal character-chopping algorithms before passing features to a localized language network. As the oldest open-source engine, Tesseract consistently exhibited the lowest recognition accuracy across the reviewed corpus. The study by [17] reported a mere 59% accuracy on Indonesian plates, while [20] observed a low 47.90%-character accuracy on blurred Bangladeshi plates. Mechanistically, if Tesseract's segmentation module fails due to spatial distortions, non-uniform backgrounds, or irregular font tracking, the downstream network receives a fractured feature map, causing unrecoverable extraction errors. However, because it avoids heavy convolutional tensor calculations, Tesseract excels in computational efficiency. The benchmark by [8] noted that Tesseract processes images 30% faster than deep learning alternatives while consuming only a fifth of the memory, making it viable solely for extreme edge-device applications relying on low-power CPUs to process standardized Latin plates.

EasyOCR introduces a modern deep learning framework by combining a deep convolutional backbone (typically ResNet or VGG) with a Bidirectional Long Short-Term Memory (BiLSTM) sequence network and a Connectionist Temporal Classification (CTC) decoder. By replacing rigid structural segmentation with fluid convolutional feature extraction, EasyOCR provides a significant architectural upgrade. The comparative analysis by [9] found that EasyOCR achieved greater than 95% accuracy compared to Tesseract's 90% on non-standard Indian plates. Furthermore, [20] confirmed it is not only more accurate but also faster on low-resolution images processed via CUDA backends (0.114 seconds versus 0.175 seconds for Tesseract). Consequently, EasyOCR strikes an optimal latency-to-accuracy balance, delivering high character precision and adequate inference speeds for real-time IoT applications operating with entry-level GPU support.

PaddleOCR enhances this convolutional recurrent design through highly optimized, lightweight model configurations. Developed by Baidu, PaddleOCR integrates a specialized Differentiable Binarization text detector (DBnet) with a robust sequential recognition network (CRNN + CTC). This architecture excels at managing the entire spatial sequence of a text string coherently. Addressing RQ1's metric of the exact-match rate, [8] provided compelling evidence: while EasyOCR frequently excels in individual character accuracy, it struggles with full-string alignment. PaddleOCR resolves this through superior sequence alignment layers, achieving a

39% exact-match rate under clean conditions, which vastly outperformed EasyOCR (27%) and Tesseract (17%). Its robust sequential decoder renders it the premier choice for enterprise applications demanding absolute full-string accuracy, such as e-ticketing or automated tolling, provided that adequate processing hardware is available.

TrOCR represents the state-of-the-art in text recognition by completely abandoning the convolutional-recurrent pipeline in favor of a pure end-to-end Transformer architecture. It processes text as a holistic global sequence, utilizing a Vision Transformer (ViT) encoder to decompose an image into spatial patches, followed by an autoregressive language decoder. By treating text extraction globally, it circumvents explicit character segmentation entirely. The framework by [2] implemented a YOLOv8 and TrOCR pipeline, achieving an impressive 0.94% Character Error Rate (CER) during complex plate transitions. However, this attention-driven approach introduces a massive computational burden. The resource profiling by [11] highlighted a critical efficiency bottleneck on edge devices (Raspberry Pi 4): while TrOCR achieved the highest accuracy (92.6%), its inference time reached 4.98 seconds per crop compared to PaddleOCR's 840 milliseconds. Thus, TrOCR's unparalleled accuracy currently demands substantial cloud or high-tier GPU infrastructures to overcome its autoregressive latency.

C. RQ2: Engine Vulnerabilities to Script Diversity and Environmental Degradation

To answer RQ2, the performance data were cross tabulated against script complexity and environmental stressors. The underlying architectural design of each engine profoundly dictates its morphological handling capabilities and environmental resilience.

Deep learning engines consistently reach near-saturation accuracy (greater than 95%) on standardized Latin formats with uniform aspect ratios, as demonstrated by [3] on Romanian infrastructure and [16] on German datasets. Conversely, Tesseract struggles significantly with non-standard Asian Latin fonts; studies [18] and [19] evaluated Tesseract on Malaysian plates, obtaining drastically lower performance metrics due to local font variations that disrupted classical line-finding algorithms. For complex cursive and tonal scripts like Bangla and Arabic, traditional segmentation logic collapses completely due to continuous horizontal strokes or contextual shape-shifting characters. Conversely, [20] and [21] showed EasyOCR maintaining up to 94% accuracy on Bangla scripts, though recognition rates dropped during high frame-rate streams. For cursive scripts, [22] successfully utilized EasyOCR combined with post-OCR SVM/KNN classifiers to achieve a perfect 100% precision on Arabic plates by algorithmically correcting CTC connectivity errors.

In the context of Devanagari and Thai scripts, dense stroke configurations and multi-tiered vertical structures present severe challenges. The evaluation by [23] tested EasyOCR on Indian plates mixing English and Marathi, achieving 88.14% accuracy. Due to non-standard plate formats, [24] reported that despite excellent bounding-box detection, downstream character recognition capped at 80%. For multi-tonal Thai characters, [25] achieved a 95.5%-character accuracy using EasyOCR. However, [26] warned that environmental deployment variables, such as camera angles deviating from a perpendicular axis, distort these vertical aspect ratios and severely degrade convolutional recognition metrics. For the most complex layouts featuring irregular spacing, such as Chinese Han plates or multilingual CIS region plates, [5] and [6] demonstrated that TrOCR's global attention mechanism seamlessly bypasses segmentation failures, achieving the lowest CER globally. Furthermore, [27] expanded this framework to multi-line sequences, proving that Transformer architectures successfully eliminate cumulative row-segmentation errors characteristic of legacy models.

Environmental stressors such as fog, rain, and motion blur universally alter pixel gradients and degrade ALPR performance, mandating adaptive preprocessing pipelines. Addressing this vulnerability, [28] conducted rigorous contrast experiments on PaddleOCR, discovering that bilateral filtering yielded the lowest average edit distance by reducing high-frequency background noise while preserving critical character boundaries. Simultaneously, Otsu thresholding maximized raw character accuracy by dynamically binarizing bimodal pixel distributions. Similarly, [29] established that motion deblurring quality directly dictates downstream Levenshtein Distance accuracy; restoring sharp character edges prevents convolutional filters from misinterpreting closely shaped alphanumeric characters. Crucially, the synthesis indicates that deep learning architectures are highly sensitive to input quality, meaning dynamic, environment-aware preprocessing is a mandatory prerequisite to maintaining their superior accuracy in unconstrained field operations.

To overcome the architectural limitations of independent detection and extraction stages, the field of ALPR is rapidly shifting toward next-generation multimodal foundation models. Conventional systems inherently suffer from cumulative pipeline errors; if the bounding-box detector fails, the downstream OCR receives fractured data. Emerging research frontiers are addressing this through text-spotting Vision Transformers and Large Vision-Language Models (LVLMs), such as Florence-2 and LLaVA [30], [31]. By processing the raw vehicle image through unified, high-dimensional attention layers, these multimodal architectures perform spatial coordinate grounding and alphanumeric sequence generation simultaneously in a single forward pass. Recent evaluations of these foundation models demonstrate human-like contextual reasoning capable of decoding severely damaged, occluded, or multi-script text across unconstrained environments without requiring task-specific fine-tuning [30].

D. RQ3: Contextual Deployment Recommendations and Taxonomy

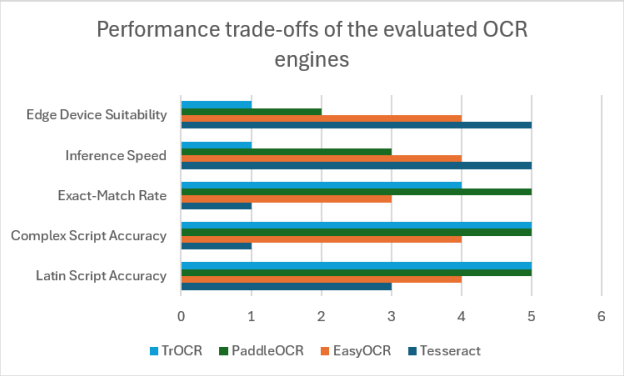

Figure 2. Radar chart illustrating the performance trade-offs of the evaluated OCR engines

To address RQ3, the computational trade-offs identified across the analyzed literature were synthesized into a decision-support framework. The visual representation in Figure 2 illustrates the multi-dimensional performance profile of the four OCR engines. It is evident that the performance landscape is governed by an inverse relationship between architectural complexity and computational velocity. While engines like TrOCR and PaddleOCR dominate in precision and script-robustness, their high-dimensional parameter space necessitates substantial hardware overhead in the form of VRAM footprint and GPU acceleration. Conversely, engines like Tesseract and EasyOCR prioritize a lightweight computing envelope, functioning viably on CPU-bound or low-tier edge nodes at the direct expense of categorical exact-match accuracy.

This multi-dimensional trade-off distinguishes this proposed taxonomy from prior generalized reviews. Rather than relying on singular accuracy metrics, developers must execute a hierarchical selection approach based on strict environmental computing envelopes. System architects must first define their application's non-negotiable hardware constraints—whether it be the strict real-time latency limitations of a microcontroller-based parking gate or the high-fidelity accuracy required for enterprise-grade electronic law enforcement on cloud servers. Furthermore, the selection process must be dynamically adjusted for script morphology; while Latin-script environments achieve high performance across all deep learning engines, complex non-Latin or multi-tonal scripts demand the feature-extraction depth provided exclusively by deep learning-based sequential decoders or global attention mechanisms. The comprehensive mapping of these architectural alignments against diverse operational environments and script complexities is systematically synthesized in Table 3.

TABLE 3.
SYNTHESIZED QUALITATIVE MATRIX OF OCR PERFORMANCE BY SCRIPT TYPE

Script Type	Tesseract	EasyOCR	PaddleOCR	TrOCR
Latin (Standardized, EU)	Moderate (78–90%)	High (>90%)	Very High (>95%)	Very High (>95%)
Latin (Non-standard, Asia)	Low (50–70%)	Mod-High (70–85%)	High (85–95%)	High (85–95%)
Bangla	Very Low (47–50%)	Mod-High (76–94%)	N/A	N/A
Devanagari (India)	Moderate (80–90%)	High (88–95%)	Very High (>95%)	N/A
Thai	N/A	High (95%)	N/A	N/A
Arabic / Iraq	Moderate (Unreported)	Very High (with post-classification)	N/A	N/A
Han (China)	N/A	N/A	N/A	Very High (CER <1%)
Multilingual (CIS)	N/A	N/A	N/A	Optimal (Lowest CER)

In summary, the synthesis confirms that no single OCR engine is universally superior across all computing boundaries. Tesseract remains strictly recommended for extreme low-resource edge devices operating under highly constrained CPU environments with standardized Latin plates, where absolute sequence accuracy is a secondary priority. EasyOCR offers the optimal accuracy-to-speed ratio, making it highly recommended for dynamic, real-time IoT applications equipped with entry-level GPU acceleration handling moderately complex non-Latin scripts. PaddleOCR possesses the most robust sequential decoder, rendering it the primary choice for applications demanding exact-match, full-string plate accuracy, provided that adequate VRAM hardware and optimal spatial preprocessing pipelines are actively maintained. Finally, TrOCR represents the definitive frontier for highly complex, multi-line, irregularly spaced, or multilingual plates; however, due to its severe autoregressive latency bottlenecks, its deployment is currently recommended exclusively for cloud-based or heavily GPU-accelerated batch-processing infrastructures. By mapping deep neural architectures against script diversity and resource availability, this systematic review successfully transcends raw accuracy reporting, providing practitioners with the precise actionable insights required to design globally resilient Intelligent Transportation Systems.

IV. CONCLUSION

This SLR conclusively demonstrates that the optimal selection of an OCR engine for ALPR systems is not a matter of absolute algorithmic superiority, but rather a strategic alignment with contextual deployment constraints. The comparative synthesis of Tesseract, EasyOCR, PaddleOCR, and TrOCR reveals distinct operational frontiers driven by the underlying architecture of each engine. While traditional segmentation-based models like Tesseract are increasingly obsolete for complex, non-standard scripts, they retain a niche viability strictly within extreme low-resource environments processing uniform Latin characters. Conversely, CNN-based architectures have effectively bridged the gap between high accuracy and edge-device viability. EasyOCR stands out as the optimal solution for dynamic, real-time IoT applications requiring a balance of speed and multi-script adaptability, whereas PaddleOCR provides unparalleled exact-match precision, making it the premier choice for high-stakes sequential recognition.

Furthermore, the integration of sequence-generating Transformer architectures, specifically TrOCR, marks a profound paradigm shift in the field. By treating text extraction as a holistic global sequence, Transformers successfully bypass the inherent limitations of character segmentation, offering a definitive solution for the most intricate multilingual, multi-line, and irregularly spaced license plates. However, this architectural supremacy is currently hindered by substantial computational latency, confining its immediate applicability to GPU-accelerated or cloud-based infrastructures rather than real-time edge processing. Beyond engine architecture, this review underscores a universal vulnerability: environmental degradation. The empirical evidence confirms that algorithmic resilience cannot be achieved through static filters; dynamic, environment-aware image preprocessing pipelines are a mandatory prerequisite for maximizing the performance of deep learning-based OCRs.

To advance the field toward universally robust Intelligent Transportation Systems, future research must prioritize three critical agendas. First, the architectural compression and optimization of heavy Transformer models (such as TrOCR) to enable real-time, lightweight edge computing without compromising accuracy. Second, the development of adaptive preprocessing algorithms capable of dynamically responding to real-time weather and lighting anomalies. Finally, the democratization of open-source, multi-country license plate datasets to eliminate regional evaluation biases. By transitioning from trial-and-error methodologies to the context-driven architectural mapping provided in this review, developers and policymakers can construct ALPR systems that are computationally efficient and resilient across the globe's highly diverse linguistic and infrastructural landscapes.

REFERENCES

[1] M. Ashkanani, "A Self-Adaptive Traffic Signal System Integrating Real-Time Vehicle Detection and License Plate Recognition for Enhanced Traffic Management," *Inventions*, vol. 10, p. 14, 2025, doi: 10.3390/inventions10010014.

[2] E. Julianto, "Automatic Plate Number Recognition (APNR) using Deep Learning with YOLOv8 and TrOCR," 2026, doi: 10.1109/ISIBER68248.2026.11470508.

[3] B. Buleu, "A Deep Learning-Based System for Automatic License Plate Recognition Using YOLOv12 and PaddleOCR," *Applied Sciences*, vol. 15, p. 7833, 2025, doi: 10.3390/app15147833.

[4] R. N. Suharto, "Comparative Evaluation of YOLO-based Models for Indonesian License Plate Detection and Recognition with EASY OCR," *Procedia Comput. Sci.*, vol. 269, pp. 943–952, 2025, doi: 10.1016/j.procs.2025.09.037.

[5] Z. Ke, "TrOCR-based single/double Chinese license plates recognition," *International Conference on Optics, Electronics, and Communication Engineering (OECE 2025)*, p. 88, 2025, doi: 10.1117/12.3090329.

[6] I. Saitov, "CIS Multilingual License Plate Detection and Recognition Based on Convolutional and Transformer Neural Networks," *Procedia Comput. Sci.*, vol. 229, pp. 149–157, 2023, doi: 10.1016/j.procs.2023.12.016.

[7] M. J. Nayeem, "An Efficient Approach to Recognize Bangla License Plate for Diverse-Quality Images," 2025, doi: 10.1007/978-981-96-3420-0_13.

[8] V. R. Thenisius, "Evaluating License Plate Recognition in Indonesia: A Comparison of OCRs on YOLOv8-Detected Plates," 2025, doi: 10.1109/ICIC68054.2025.11308159.

[9] D. R. Vedhaviyassh, "Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm," 2022, doi: 10.1109/ICECA55336.2022.10009215.

[10] P. P. Reddy, "License Plate Detection using YOLO v8 and Performance Evaluation of EasyOCR, PaddleOCR and Tesseract," 2024, doi: 10.1109/ICCCNT61001.2024.10725878.

[11] N. Shabbir, "Comparative Performance and Resource Utilization Analysis of OCR Models for Number Plate Recognition on Raspberry Pi 4," 2025, doi: 10.1109/ComTech65062.2025.11034455.

[12] G. V. Reddy, "A comprehensive review on helmet detection and number plate recognition approaches," *E3S Web of Conferences*, vol. 507, p. 1075, 2024, doi: 10.1051/e3sconf/202450701075.

[13] F. Delgado, S. Garrido, and B. S. Bezerra, "Barriers to Visibility in Supply Chains: Challenges and Opportunities of Artificial Intelligence Driven by Industry 4.0 Technologies," *Sustainability*, vol. 17, p. 2998, 2025, doi: https://doi.org/10.3390/su17072998.

[14] M. Pilliang, Munawar, M. A. Hadi, G. Firmansyah, and B. Tjahjono, "Predicting Risk Matrix in Software Development Projects using BERT and K-Means," in *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Jakarta, Indonesia: IEEE, Oct. 2022, pp. 137–142. doi: 10.23919/EECSI56542.2022.9946637.

[15] Z. Munn, S. Moola, D. Riitano, and K. Lisy, "The development of a critical appraisal tool for use in systematic reviews addressing questions of prevalence," *Int. J. Health Policy Manag.*, vol. 3, no. 3, pp. 123–128, 2014, doi: 10.15171/ijhpm.2014.71.

[16] H. Aljzaere, "German License Plate Recognition System Using the YoloV8 Model and EasyOCR," 2024, doi: 10.1109/ICSJ62869.2024.10804754.

[17] N. R. Adytia and G. P. Kusuma, "Indonesian license plate detection and identification using deep learning," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 7, pp. 1–7, Jul. 2021, doi: 10.46338/ijetae0721_01.

[18] M. L. Tham, "IoT Based License Plate Recognition System Using Deep Learning and OpenVINO," *2021 4th International Conference on Sensors, Signal and Image Processing*, pp. 7–14, 2021, doi: 10.1145/3502814.3502816.

[19] R. Parvathi, "Automated Vehicle Number Plate Detection Using Tesseract and Paddleocr," *Advances in Computational Intelligence and Robotics*, pp. 90–107, 2023, doi: 10.4018/978-1-6684-9189-8.ch007.

[20] M. J. Nayeem, "Performance Evaluation of Tesseract and EasyOCR to Recognize Bangla License Plate for Diverse Quality Image," 2024, doi: 10.1109/ICCIT64611.2024.11021754.

[21] M. H. Tusar, "Real Time Bangla License Plate Recognition with Deep Learning Techniques," 2022, doi: 10.1109/IICAIET55139.2022.9936764.

[22] N. M. I. Abdul-Sattar, "AN INTELLIGENT SYSTEM FOR IRAQI ARABIC LICENSE PLATE RECOGNITION USING YOLO AND MACHINE LEARNING," *Kufa Journal of Engineering*, vol. 17, pp. 417–436, 2026, doi: 10.30572/2018/KJE/170225.

[23] H. U. Iyer, "Automatic Number Plate and Face Recognition System for Secure Gate Entry into Military Establishments," 2024, doi: 10.1109/ICSSES62373.2024.10561368.

[24] S. Rani, "Recognition of High-Security Registration Plate for Indian Vehicles," 2023, doi: 10.1109/AISP57993.2023.10134888.

[25] V. Phon, "Integrating YOLOv5 and EasyOCR for Robust Thai License Plate Recognition," 2025, doi: 10.1109/ECTI-CON64996.2025.11101195.

[26] P. Netinant, "Evaluating Factors Shaping Real-Time Internet-of-Things-Based License Plate Recognition Using Single-Board Computer Technology," *Technologies (Basel).*, vol. 12, p. 98, 2024, doi: 10.3390/technologies12070098.

[27] H. Lin, "An End-to-End Recognition Model for Multi-Line Characters of Ship Plates Based on TrOCR," 2025, doi: 10.1109/HPCC67675.2025.00238.

[28] G. A. Hakasin, "Impact of image pre-processing on enhancing paddleOCR performance for automatic indonesia number plate recognition," *Procedia Comput. Sci.*, vol. 269, pp. 1299–1308, 2025, doi: 10.1016/j.procs.2025.09.071.

[29] F. Karadag, "Reading Between the Blurs: A Comparative Analysis of Motion Deblurring Methods for Vehicle License Plate OCR," *Procedia Comput. Sci.*, vol. 269, pp. 1712–1722, 2025, doi: 10.1016/j.procs.2025.09.114.

[30] B. Xiao *et al.*, "Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 4818–4829. doi: 10.1109/CVPR52733.2024.00461.

[31] Y. Liu *et al.*, "OCRBench: on the hidden mystery of OCR in large multimodal models," *Science China Information Sciences*, vol. 67, no. 12, Dec. 2024, doi: 10.1007/s11432-024-4235-6.

APPENDIX A. SUMMARY OF PRIMARY STUDIES

No	Ref	Authors / Year	Country / Region	Script Type(s)	OCR Engine(s) Evaluated	Detection Model	Dataset Characteristics	Key Performance Findings
1	[2]	Julianto (2026)	Indonesia	Latin (mixed, non-standard)	TrOCR	YOLOv8	Real-world vehicle captures	Achieved 0.94% Character Error Rate (CER) during complex plate transitions, demonstrating Transformer superiority for challenging layouts.
2	[3]	Buleu (2025)	Romania	Latin (standardized, EU)	PaddleOCR	YOLOv12	Standardized Romanian infrastructure dataset	Demonstrated very high (>95%) recognition accuracy on uniform European plate formats.
3	[5]	Ke (2025)	China	Han (Chinese)	TrOCR	None (end-to-end)	Single/double Chinese license plate datasets	Achieved CER < 1%, effectively eliminating segmentation errors for complex, irregularly spaced Han characters.
4	[6]	Saitov (2023)	CIS Region	Multilingual (Cyrillic, mixed)	TrOCR	CNN + Transforme r (hybrid)	Multilingual CIS plate dataset	Attained the lowest CER globally across all evaluated engines, proving robust for cross-script alphabets.
5	[8]	Thenisius (2025)	Indonesia	Latin (non-standard, Asian)	Tesseract, EasyOCR, PaddleOCR	YOLOv8	500 images of Indonesian plates	PaddleOCR achieved highest Exact-Match Rate (EMR) at 39%; EasyOCR at 27%; Tesseract at 17%. Tesseract was 30% faster but only 59% accurate.
6	[9]	Vedhaviy assh (2022)	India	Latin (Indian non-standard)	Tesseract vs. EasyOCR	Deep Learning (CNN)	Non-standard Indian plates	EasyOCR achieved > 95% accuracy, significantly outperforming Tesseract (~90%) due to CNN-based feature extraction.
7	[10]	Reddy (2024)	India	Latin (non-standard)	Tesseract, EasyOCR, PaddleOCR	YOLOv8	General Indian vehicle plates	Comparative evaluation confirmed deep learning engines (EasyOCR, PaddleOCR) consistently surpassed Tesseract in complex script environments.
8	[11]	Shabbir (2025)	General (Simulated)	Latin (standardized test)	Tesseract, EasyOCR, PaddleOCR, TrOCR	None (pure OCR inference)	Standardized benchmark crops	TrOCR achieved highest accuracy (92.6%) but massive latency (4.98 sec/crop). PaddleOCR balanced at 840 ms. Tesseract was fastest but least accurate.
9	[16]	Aljzaere (2024)	Germany	Latin (standardized, EU)	EasyOCR	YOLOv8	German license plate dataset	Achieved high recognition accuracy (>95%) on standardized EU plates with uniform aspect ratios.

10	[17]	Adytia & Kusuma (2021)	Indonesia	Latin (non-standard)	Tesseract	Deep Learning	Indonesian plate dataset	Reported only 59% accuracy, highlighting Tesseract's fragility against non-standard Asian Latin fonts and background noise.
11	[18]	Tham (2021)	Malaysia	Latin (non-standard)	Tesseract	YOLO + OpenVINO	Malaysian plates (local font variations)	Performance dropped drastically due to local font variations that disrupted Tesseract's classical line-finding and chopping algorithms.
12	[19]	Parvathi (2023)	Malaysia	Latin (non-standard)	Tesseract vs. PaddleOCR	Not specified	Malaysian vehicle plates	PaddleOCR demonstrated superior robustness over Tesseract for non-standard local fonts, validating CNN-sequence decoders.
13	[20]	Nayeem (2024)	Bangladesh	Bangla (non-Latin, cursive)	Tesseract vs. EasyOCR	Not specified	Bangladeshi plates (diverse quality)	EasyOCR maintained up to 94% accuracy; Tesseract plum meted to 47.9% due to continuous horizontal strokes breaking segmentation logic.
14	[21]	Tusar (2022)	Bangladesh	Bangla (non-Latin)	EasyOCR	Deep Learning	Real-world Bangla plates	Confirmed EasyOCR's high recognition rates for Bangla scripts, though rates dropped during high frame-rate streams.
15	[22]	Abdul-Sattar (2026)	Iraq	Arabic (cursive, contextual)	EasyOCR (+SVM/KNN post-correction)	YOLO	Iraqi Arabic plates	Achieved a perfect 100% precision by algorithmically correcting CTC connectivity errors using post-OCR SVM/KNN classifiers.
16	[23]	Iyer (2024)	India	Devanagari / Marathi (multi-tiered)	EasyOCR	Not specified	Indian plates (English + Marathi mix)	Achieved 88.14% accuracy; dense stroke configurations and multi-tiered vertical structures posed moderate challenges.
17	[25]	Phon (2025)	Thailand	Thai (multi-tonal, dense)	EasyOCR	YOLOv5	Thai license plates	Attained 95.5%-character accuracy, demonstrating strong performance for complex tonal scripts when coupled with robust detection.
18	[26]	Netinant (2024)	Thailand	Thai (multi-tonal)	EasyOCR	Not specified	Real-time IoT / Edge deployment	Warned that camera angles deviating from perpendicular severely degrade convolutional metrics, distorting vertical aspect ratios.
19	[27]	Lin (2025)	General (Maritime)	Multi-line / Irregular spacing	TrOCR	None (end-to-end)	Ship plate multi-line datasets	Proved Transformer architectures successfully eliminate cumulative row-segmentation errors characteristic of legacy CNN-RNN models.
20	[28]	Hakasin (2025)	Indonesia	Latin (noisy/ degraded)	PaddleOCR	Not specified	1000+ images with fog/background noise	Bilateral filtering yielded the lowest average edit distance; Otsu thresholding maximized raw character accuracy by dynamically binarizing pixel distributions.
21	[29]	Karadag (2025)	General (Simulated)	Latin (motion blur)	Various (implicitly Tesseract, EasyOCR, PaddleOCR, TrOCR)	Not specified	Motion-blurred plate crops	Established that deblurring quality directly dictates downstream Levenshtein Distance accuracy; restoring sharp edges prevents convolutional filters from misinterpreting characters.